

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A database engine comprising:
a page aggregator component that operates across concurrent transactions to obtain information on an aggregate size change that occurs on a data page; the data page is copied by transaction(s) that requires modification thereof.
2. (Original) The database engine of claim 1 further comprising a heap allocation component that employs the information to determine an availability of space for the data page.
3. (Previously Presented) The database engine of claim 2, further comprising a lock manager that enables sub page level locking across concurrent transactions.
4. (Previously Presented) The database engine of claim 3, each of the concurrent transactions modifies a respective copy of the data page.
5. (Previously Presented) The database engine of claim 3, the page aggregator component enables a determination of space consumptions across a respective copy of data page employed by each transaction.
6. (Previously Presented) The database engine of claim 4, the page aggregator component determines the space consumption across the respective copy from information available in the lock manager.

7. (Previously Presented) The database engine of claim 3, the sub page level_locking is a row level locking.
8. (Previously Presented) The database engine of claim 2, the heap allocation component tracks an availability of space on the data page.
9. (Previously Presented) The database engine of claim 2, the heap allocation component and the page aggregator component enforce a set of conditions on a transaction that operates on the data page, such that a space availability for the data page prior to the commit stage of the transaction is assured.
10. (Previously Presented) A database engine comprising:
 - a page aggregator that works across concurrent transactions that perform sub-page level operations on a data page; and
 - a heap manager that tracks a space availability for the data page *via* information supplied by the page aggregator.
11. (Original) The data base engine of claim 10, each of the concurrent transactions modify a respective copy of the data page without consuming all space available on the data page.
12. (Original) The data base engine of claim 10, further comprising a lock manager that facilitate compatibility of operations across the concurrent transactions.
13. (Original) A method that facilitates synchronization in concurrent transactions comprising:
 - obtaining information on an aggregate size change that occur on a data page as a result of concurrent transactions operating on respective copies of the data page; and
 - tracking a space availability for the data page over all the concurrent transactions.

14. (Original) The method of claim 13 further comprising assigning locks to a resource on the data page.
15. (Original) The method of claim 13 further comprising modifying the data page.
16. (Original) The method of claim 13 further comprising replacing a row with an inserting pointer.
17. (Original) The method of claim 16 further comprising inserting the row on a new page.
18. (Original) The method of claim 14 further comprising storing the information in the locks.
19. (Original) The method of claim 18 further comprising discarding the locks upon a roll back of a transaction.
20. (Original) The method of claim 18 further comprising discarding the locks upon committing a transaction.
21. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 13.
22. (Original) A system for facilitating synchronization in concurrent transactions comprising:
 - means for obtaining information on an aggregate size change that occur on a data page during modifications thereof by concurrent transactions; and
 - means for tracking a space availability for the data page over all the concurrent transactions.

23. (Original) The system of claim 22 further comprising means for assuring availability of space on a data page prior to a commit stage of a transaction operating on the data page.

24. (Original) A computer-readable medium having stored thereon a data structure comprising:

a computer executable component that operates across a plurality of active transactions to obtain information on a space change for a data page, the data page being copied by an active transaction that requires a modification thereof.

25. (Original) The computer readable medium of claim 24 further comprising a further computer executable component that tracks an availability of space for the data page based on the information.

26. (Previously Presented) The computer readable medium of claim 24 further comprising an additional computer executable component that administers logical considerations during data modification on the data page.

27. (Original) The computer readable medium of claim 26 the additional computer executable component grant locks at a row level of the data page.

28. (Original) The computer readable medium of claim 27 further comprising a forwarding pointer device that guides a query to a new location of data.

29. (Original) A system for facilitating synchronization in concurrent transactions comprising:

means for determining logical permission to insert a row on a data page during modifications thereof by concurrent transactions; and

means for determining a space availability for the data page over the concurrent transactions.

30. (Original) The system of claim 29 further comprising means for mitigating reorganization of data around the data page at the commit stage of the concurrent transactions.
31. (Original) A method for manipulating data in a data page by a transaction comprising:
 copying a data page to a reserved space for the transaction; and
 determining an aggregate size change for the data page.
32. (Original) The method of claim 31 further comprising tracking a space availability on the data page across a plurality of concurrent transactions working on the data page.
33. (Original) The method of claim 31 further comprising creating a new page and inserting a row therein.
34. (Original) The method of claim 33 further comprising employing a pointer in the data page to guide a query to the row in the new page.
35. (Original) The method of claim 31 further comprising locking a resource at a row level on the data page.
36. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 31.